IPMI v1.5 Overview

IPMI defines common interfaces to the “intelligent” hardware that is used to monitor server physical health characteristics, such as temperature, voltage, fans, power supplies, and chassis intrusion. In addition to health monitoring, IPMI includes other system management capabilities that help drive down the total cost of ownership (TCO) including automatic alerting, automatic system shutdown and restart, remote restart and power control capabilities, and asset tracking.

New remote management interfaces in IPMI v1.5 facilitate the management of rack mount Internet servers and systems in remote environments. IPMI v1.5 also includes extensions to support other existing and emergent standards such as DMTF* Alert Standard Forum, CompactPCI®*, SMBus* 2.0, and PCI* Management Bus.

IT Benefits from Standard, Manageable Server Platforms -

IPMI-based server management reduces TCO by allowing IT managers to determine the health of their server hardware, whether the server is running normally or is in a non-operational state. Servers based on IPMI use “intelligent” or autonomous hardware that remains operational even when the processor is down so that platform management information and control capabilities are always accessible. The robust, efficient, and authenticated IPMI interfaces enable access to the same management capabilities from Serial/Modem, LAN, local management software, third party emergency management add-in cards, and other IPMI-enabled servers under ALL system phases: power-down, pre-boot, OS load and run-time.

New automatic alerting and recovery actions in IPMI v1.5 allow remote IT personnel to be notified when a problem occurs. Alerts can even be sent to multiple LAN and serial/modem destinations. Recovery actions include the ability to remote command a system to power on/off, power cycle, reset, or trigger a diagnostic interrupt. This includes the capability to set ‘boot options’ that direct the booting of the system after a remotely initiated startup. For example, a remote administrator could initiate a remote reset and direct the system to boot to a ‘service partition’ instead of the main OS partition.

IT managers gain flexible and interoperable access to vital platform management information. System-to-system monitoring or management via a connected server is becoming increasingly important as IT managers deploy complex system topologies such as clusters and rack-mounted configurations. In addition, the scalable nature of IPMI enables the architecture to be deployed across a server product line, from entry to high-end servers, giving IT managers a consistent base of platform management functionality upon which to effectively manage their servers.

Broad Industry Adoption -

Since IPMI v1.0 was introduced in 1998, over 70 companies representing a broad cross-section of the industry have adopted IPMI (including system and motherboard OEMs, silicon vendors, and embedded computer manufacturers). Several companies are also creating baseboard and peripheral management controllers for IPMI including Agilent* Technologies, QLogic* Corporation, National Semiconductor*, Vitesse* Semiconductor, and Winbond* Electronics Corporation. In addition, the vendors are including IPMI firmware and SDKs in their offerings. The IBM-AIX* and HP-

* Third party marks and brands are property of their respective holders.

UX* operating systems are providing IPMI messaging support, and support is under discussion with other Operating System Vendors as well. For a complete list of IPMI adopters visit:
http://developer.intel.com/design/servers/ipmi/adopterlist.htm

OEMs See R&D Savings, Faster Time to Market -

IPMI v1.5's common interfaces for remote management have enabled the creation of 3rd party hardware, firmware, and software building blocks. Together building blocks enable OEMs to efficiently develop and deploy interoperable remote management solutions. The scalability and commonality of the interfaces also enables a high degree of hardware, software, and firmware re-use across a product line.

For example, the same IPMI messages that are used by local software for accessing system health sensor, event log, and recovery features can be delivered via the new serial and LAN interfaces. This enables an 'in-band' local management software stack to be rapidly converted to an 'out-of-band' remote software stack by changing the underlying communication 'driver' used to access the system.

Extensible Framework for Differentiation and Future Features -

IPMI v1.5 carries forward the IPMI v1.0 philosophy of supporting OEM differentiation and feature extension on top of the IPMI framework. This includes the ability to support special sensors, events, commands, data records, and control capabilities. IPMI v1.5 extends this to allow for OEM communication interfaces, automated actions, and alert data. The IPMI messaging framework that supports OEM communication capabilities also enables the specification to be readily extended to support new communication media. For example, future support for delivering IPMI messages via an InfiniBand™ communication interface is under investigation.

Technology Foundation -

The 'Intelligence' in IPMI comes from a management micro-controller. For a host system, this controller is called the BMC (Baseboard Management Controller). The BMC operates on standby power and autonomously polls system health status. If it sees any elements go out-of-range, it can take actions such as logging the event, generating alerts, and event performing automatic recovery actions such as system power down or resets. Associated with the BMC is a set of non-volatile storage that holds the Sensor Data Records (SDRs), System Event Log, and Field Replaceable Unit information (e.g. serial number and part number information that is used to identify different serviceable or failed entities in a system).

IPMI works by specifying common, abstracted, message-based interfaces to the management micro-controller. This abstraction isolates software from the hardware implementation. In addition, IPMI specifies commands and a set of "Sensor Data Records" that describe the number and type of monitoring and control capabilities offered by a given platform. These allow software to discover and automatically adapt to the monitoring and control features offered by each platform.

IPMI v1.5 builds on the proven technology from IPMI v1.0 and exposes these same capabilities through the new interfaces, allowing both local and remote software to automatically configure itself to multiple systems. This facilitates the creation of cross-platform management software.

Backward Compatibility -

One IPMI 1.5 goal was to retain backward compatibility with IPMI v1.0 while adding major new capabilities to the specification. Sensor operation, sensor data records, system event log, FRU, watchdog timer, and system interfaces are almost completely unchanged from IPMI v1.0. Thus, the firmware and software for a system that essentially provides the same features as a v1.0 system will require very few modifications to be IPMI v1.5 conformant. It is straightforward to create software that can support both IPMI v1.5 and v1.0 based systems. Firmware and software will, of course,

need to be modified and extended to take advantage of any of the new capabilities that IPMI v1.5 systems offer.

New Capabilities in IPMI v1.5 -

Many significant capabilities have been added to the specification, including:

IPMI over LAN	Using IPMI messages encapsulated in UDP datagrams. The packet format follows the DMTF Pre-OS Working Group's RMCP UDP Packet Format
IPMI over Serial/Modem	Including three specified protocol modes providing different tradeoffs of efficiency and standardization: • Basic Mode for highest speed with automated remote consoles • PPP Mode for support of widely available communication stacks • Terminal Mode for limited 'dumb terminal' text access in legacy environments
Platform Event Filtering (PEF)	PEF is the ability to generate selectable actions when a new event matches up to a configurable set of 'event filters'. Actions include power off, power cycle, reset, and send alert. The configurable event filters include a 'wildcarding' capability that enables a filter to range from very generic to very specific. For example, you could configure one filter to generate an action on any 'non-critical' event, and another filter to only generate an action on
LAN Alerting	LAN alerts are sent as SNMP Traps in the PET (Platform Event Trap) format to a specified alert destination. SNMP Traps are commonly used as 'unreliable' datagrams that are sent out with no acknowledgement that they were received. IPMI v1.5 provides a configuration option that allows the IPMI Alert traps to be acknowledged and retried if there's no response. This can be used in conjunction with corresponding software at the alert destination to provide 'reliable' delivery of the PET.
Serial/Modem Alerting	Specification includes support for: • Dial Paging (numeric paging via a modem) • TAP Paging (alphanumeric paging via a modem connection to a TAP 1.3-based paging service) • PPP Alerting (PET Trap sent by dialing up and connecting via PPP to a remote LAN)
Alert Policies	IPMI v1.5 supports LAN and serial/modem alerts, with configurable 'Alert Policies' that support alerts to multiple destinations. The destinations in an alert policy can contain a mix of serial and LAN destinations and alert types. In addition, you can individually configure each destination to be used only if an alert to a prior destination in the policy was unsuccessful. This type of capability is also referred to as a 'call down list'. The specification allows an implementation to provide multiple Alert Policies. This enables a system to provide different policies for different classifications of events. For example, a system could be configured with one Alert Policy for 'high priority' events, and a different policy for 'low priority' events. Alert Policies are triggered by Platform Event Filtering (PEF). Each PEF event filter contains a selector that chooses which Alert Policy will be executed if that filter is matched.
Serial Port Sharing	IPMI v1.5 includes specifications managing logic that enables a single serial connector to be shared between the motherboard's serial controller and the serial connection to the management controller. This includes support for the Microsoft Whistler 'headless' service processor switching & reset escape sequences.
Boot Options	IPMI v1.5 includes the definition of a common set of flags that can be retrieved by BIOS and used to direct the system boot process. For example, the options can direct BIOS to boot to a 'service partition' instead of to the main OS partition. The boot options also include a 'mailbox' that can be used to pass special data to a BIOS or OS loader.
Callback	IPMI v1.5 supports two types of callback via a serial connection: IPMI Messaging callback, and CBCP callback. IPMI Messaging callback is initiated by issuing a 'callback' command to the BMC. This triggers the BMC to terminate the connection and call the pre-configured phone number. CBCP is the Microsoft-specified 'Callback Control Protocol'. CBCP callback is used for PPP connections only.
PCI Management Bus support	IPMI v1.5 adds commands and protocol for sending/receiving IPMI messages via the proposed PCI Management Bus. The protocols also allow a BMC to serve as the 'Host Controller' for the bus.
Users, Privileges, and Authentication Support	IPMI v1.5 includes significant support for authenticating access to the serial/modem and LAN interfaces, including: • Support for multiple users with different privilege levels and interface access rights. This includes privilege levels include Callback, User, Operator, and Administrator access. • A challenge/response protocol for authenticating users. • MD2 and MD5-based signatures on serial and LAN messages • Support for CHAP/MS-CHAP/PAP on PPP connections • Support for using Callback as an authentication mechanism for serial/modem communications

The IPMI Specifications -

IPMI is actually a set of specifications. The main document is called the Intelligent Platform Management Interface v1.5 Specification. It defines the common commands, data structures, and message formats that apply to all interfaces in IPMI. It also defines common management functions such how the System Event Log and Sensor Data Records are managed and accessed, how the system interfaces work, how sensors operate, how control functions such as system power on/off and reset are initiated, and how the IPMI host-system watchdog timer function operates.

In addition, there are several supporting documents: The Intelligent Platform Management Bus (IPMB) specification defines an internal management expansion bus that is typically used to link chassis management features with motherboard management subsystem. The IPMB Address specification specifies how devices are allocated addresses on an IPMB. The Platform Management FRU specification specifies the storage format for 'Field Replaceable Unit' information. The Platform Event Trap (PET) format specification defines the format for LAN alerts. And the Intelligent Chassis Management Bus (ICMB) specification defines a dedicated 'inter-chassis' management bus for interconnecting IPMI management between multiple host systems and peripheral chassis.

Where To Get More Information -

IPMI specifications and other information can be obtained from the IPMI web site at:
http://developer.intel.com/design/servers/ipmi

In addition to the current and prior specifications - presentations, a mail list, and FAQ information are also available, along with information on how to become an IPMI Adopter.

IPMI Adopters gain additional benefits plus software available via the web site, including development tools, drivers, early specifications, and the IPMI Conformance Test Suite.